Capstone Project-3

Supervised ML on Company Bankruptcy Prediction (Classification)

- Soma Pavan Kumar(Pro Flex)
- spkumar1998@gmail.com

Contents

1. Problem Statement
2. Introduction
3. Data Cleaning and Data Viz
4. Data Modelling and Implementation
5. Conclusion

Problem Statement

- Prediction of bankruptcy is a phenomenon of increasing interest to firms who stand to lose money because on unpaid debts. Since computers can store huge dataset pertaining to bankruptcy making accurate predictions from them before hand is becoming important.

- In this project you will use various classification algorithms on bankruptcy dataset to predict bankruptcies with satisfying accuracies long before the actual event

Introduction

- The economic meltdown of 2008, initiated a conversation about market sustainability, and the tools that can be used to predict it. The need for better predictive models become apparent, in order to avoid such devastating events in the future.

- Bankruptcy of companies and enterprises effects the financial market at multiple fronts, and hence the need to predict bankruptcy among companies by monitoring multiple variables takes on an added significance.

- A better understanding of bankruptcy and the ability to predict it will impact affect the profitability of lending institutions worldwide

Variable Information

- The Dataset we are working on has 96 Columns. Of which, 95 are variables(features) and the other is label(dependent Variable)
- The Columns are named from X1 till X95 .
- Some of the columns are:
    - X1 - ROA(C) before interest and depreciation before interest: Return On Total Assets(C)
    - X4 - Operating Gross Margin: Gross Profit/Net Sales
    - X20 - Cash Flow Per Share
    - X21 - Revenue Per Share (Yuan ¥): Sales Per Share
- And many more.....

Data Wrangling

- Data wrangling is the process of cleaning and unifying messy and complex data sets for easy access and analysis.

- Our Dataset includes about 96 columns and about 6819 observations

- Of the 96 columns only one target variable called “Bankruptcy?”

The snippet of Our dataset looks like:

```
df.head()
```

	Bankrupt?	ROA(C) before interest and depreciation before interest	ROA(A) before interest and % after tax	ROA(B) before interest and depreciation after tax	Operating Gross Margin	Realized Sales Gross Margin	Operating Profit Rate	Pre-tax net Interest Rate	After-tax net Interest Rate	Non-industry income and expenditure/revenue	Continuous interest rate (after tax)	Operat Expe R
0	1	0.370594	0.424389	0.405750	0.601457	0.601457	0.998969	0.796887	0.808809	0.302646	0.780985	1.256969e
1	1	0.464291	0.538214	0.516730	0.610235	0.610235	0.998946	0.797380	0.809301	0.303556	0.781506	2.897851e
2	1	0.426071	0.499019	0.472295	0.601450	0.601364	0.998857	0.796403	0.808388	0.302035	0.780284	2.361297e
3	1	0.399844	0.451265	0.457733	0.583541	0.583541	0.998700	0.796967	0.808966	0.303350	0.781241	1.078888e
4	1	0.465022	0.538432	0.522298	0.598783	0.598783	0.998973	0.797366	0.809304	0.303475	0.781550	7.890000e

5 rows × 96 columns

The df.head() method shows us the first 5 rows of the Dataset.

Let us look at some statistics of the Data

- The Statistics of the data could be found out from an inbuilt function in pandas library called describe() .

```
df.describe()
```

	Bankrupt?	ROA(C) before interest and depreciation before interest	ROA(A) before interest and % after tax	ROA(B) before interest and depreciation after tax	Operating Gross Margin	Realized Sales Gross Margin	Operating Profit Rate	Pre-tax net Interest Rate	After-tax net Interest Rate	Non-industry income and expenditure/revenue	Continuous interest rate (after tax)	Operating Expense Rate	Research and development expense rate	
count	6819.000000	6819.000000	6819.000000	6819.000000	6819.000000	6819.000000	6819.000000	6819.000000	6819.000000	6819.000000	6819.000000	6.819000e+03	6.819000e+03	6
mean	0.032263	0.505180	0.558625	0.553589	0.607948	0.607929	0.998755	0.797190	0.809084	0.303623	0.781381	1.995347e+09	1.950427e+09	
std	0.176710	0.060686	0.065620	0.061595	0.016934	0.016916	0.013010	0.012869	0.013601	0.011163	0.012679	3.237684e+09	2.598292e+09	
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000e+00	0.000000e+00	
25%	0.000000	0.476527	0.535543	0.527277	0.600445	0.600434	0.998969	0.797386	0.809312	0.303466	0.781567	1.566874e-04	1.281880e-04	
50%	0.000000	0.502706	0.559802	0.552278	0.605997	0.605976	0.999022	0.797464	0.809375	0.303525	0.781635	2.777589e-04	5.090000e+08	
75%	0.000000	0.535563	0.589157	0.584105	0.613914	0.613842	0.999095	0.797579	0.809469	0.303585	0.781735	4.145000e+09	3.450000e+09	
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	9.990000e+09	9.980000e+09	

8 rows × 96 columns

Checking for any NULL values in the whole Dataset

```
## Checking NULL Values
df.isna().sum()
```

```
Bankrupt?                                                  0
 ROA(C) before interest and depreciation before interest   0
 ROA(A) before interest and % after tax                    0
 ROA(B) before interest and depreciation after tax         0
 Operating Gross Margin                                    0
                                                          ..
 Liability to Equity                                       0
 Degree of Financial Leverage (DFL)                        0
 Interest Coverage Ratio (Interest expense to EBIT)        0
 Net Income Flag                                           0
 Equity to Liability                                       0
Length: 96, dtype: int64
```

- It is very evident that we don't have any null values.
- So, we can proceed with next steps

Checking for any Duplicate Values:

```
df.duplicated()
```

```
0       False
1       False
2       False
3       False
4       False
        ...
6814    False
6815    False
6816    False
6817    False
6818    False
Length: 6819, dtype: bool
```

```
[58] df.duplicated().sum()
```

```
0
```

- So, It is evident that there are no duplicate values in our whole dataset.

Lets begin with visualizations

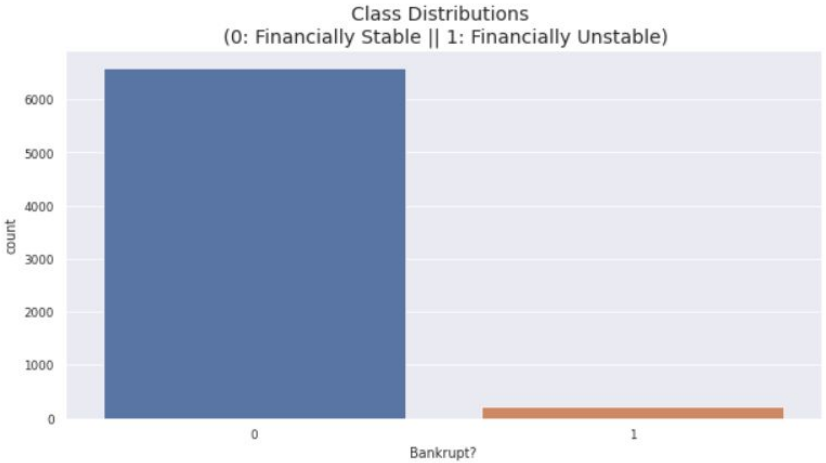
1. Checking the Distribution of the Target Variable

```
[16] print('Financially stable: ', (df['Bankrupt?'].value_counts()[0]/len(df) * 100), '% of the dataset')
```

```
Financially stable:  96.77372048687491 % of the dataset
```

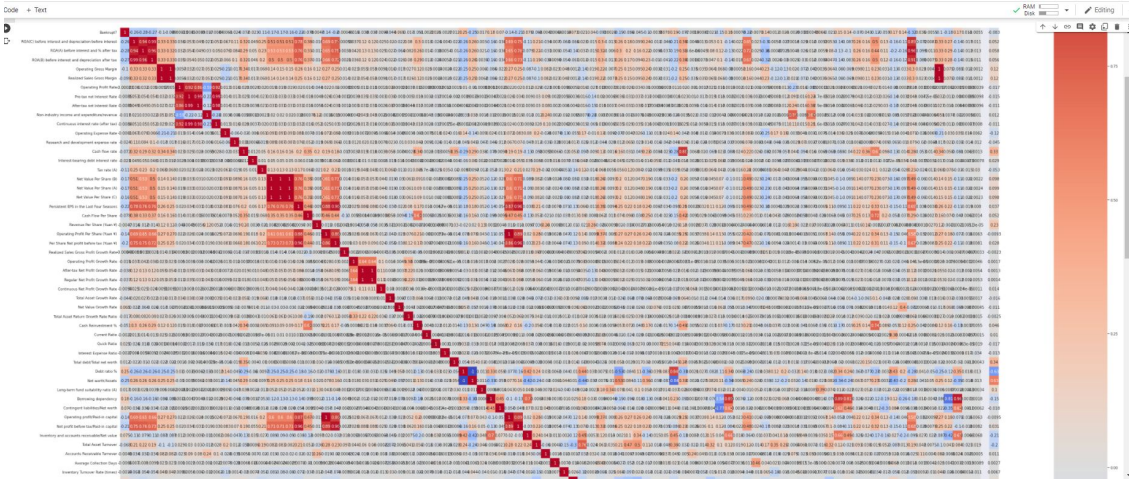
```
[17] print('Financially unstable: ', (df['Bankrupt?'].value_counts()[1]/len(df) * 100), '% of the dataset')
```

```
Financially unstable:  3.2262795131250916 % of the dataset
```



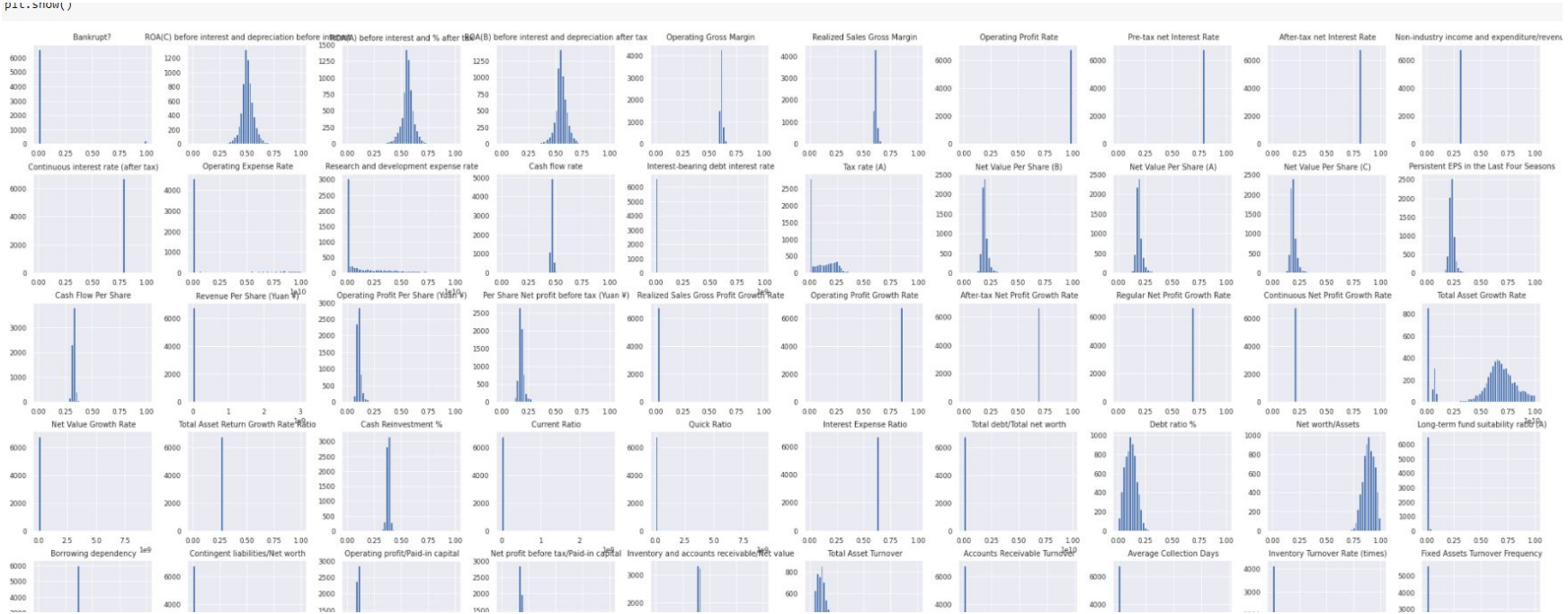
- As it can be seen, around 97% of the data (target Variable) is of Financially Stable and the other part is of Financially Unstable.

2. Lets plot the Correlations between the independent variables/features in the dataset



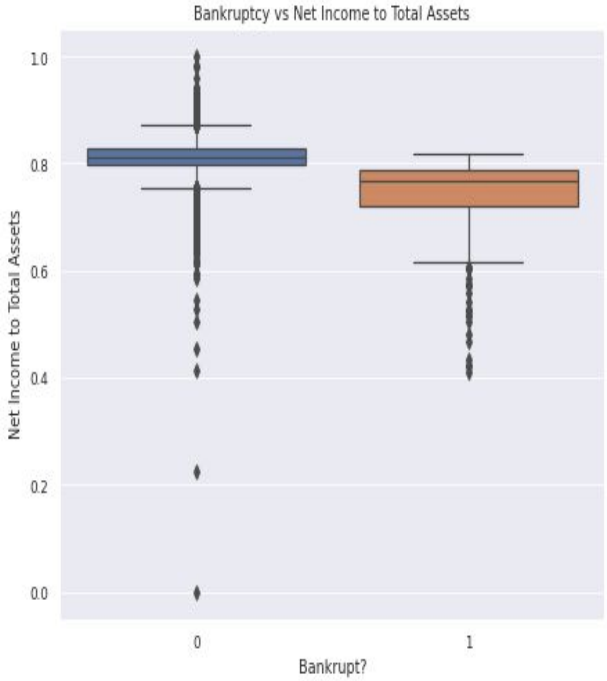
- As there are 65 features the Visualizations can get pretty difficult to understand and to show in a single slide
- Here, we cannot show all the observations, but how much ever we can see, We can see some pretty high Correlations between the Features

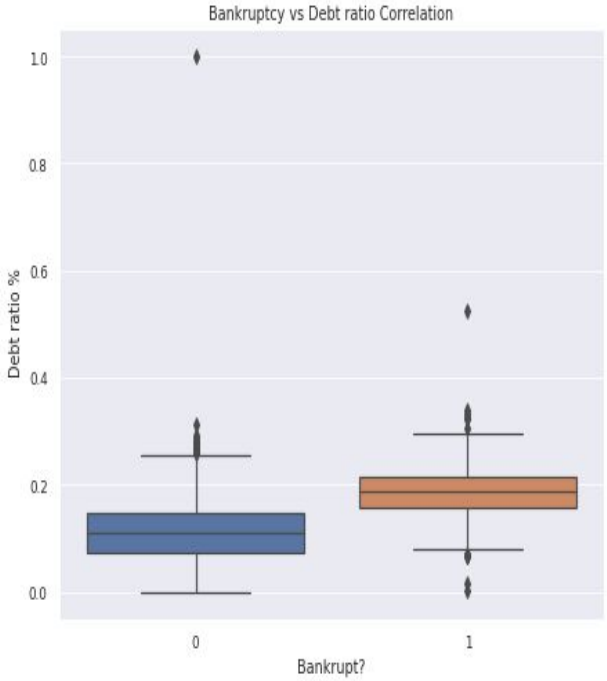
3. Data Distributions of all the variables

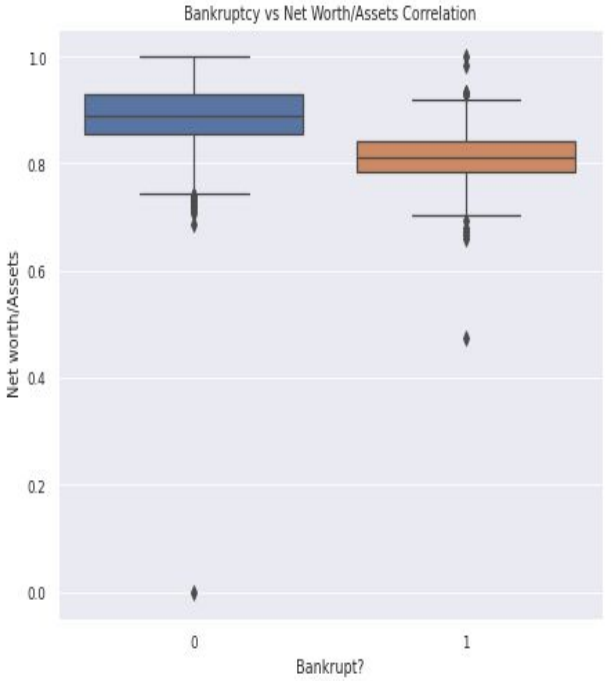


All the histograms cannot be shown in a single slide . So, these are some of the Distributions

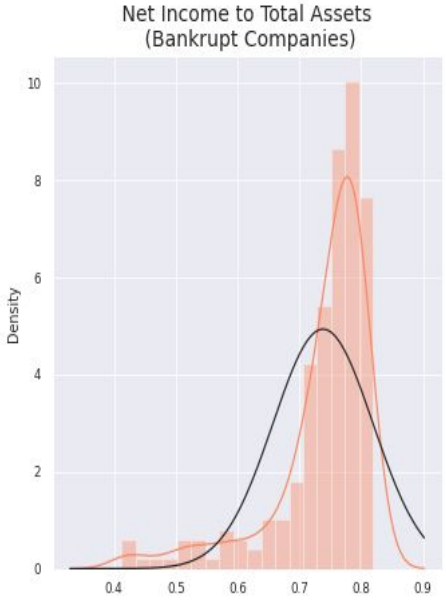
Some Correlations between the Variables using Box plot:

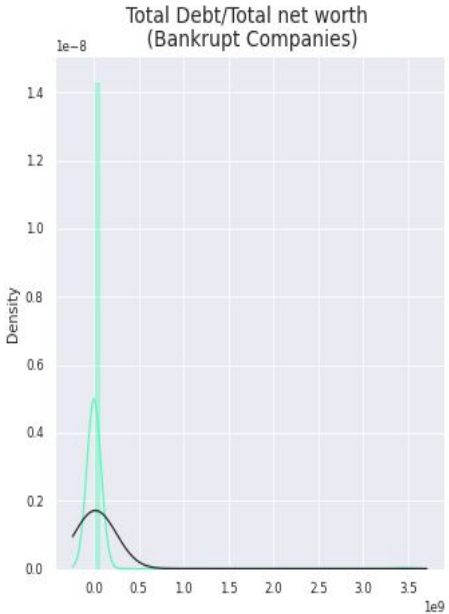


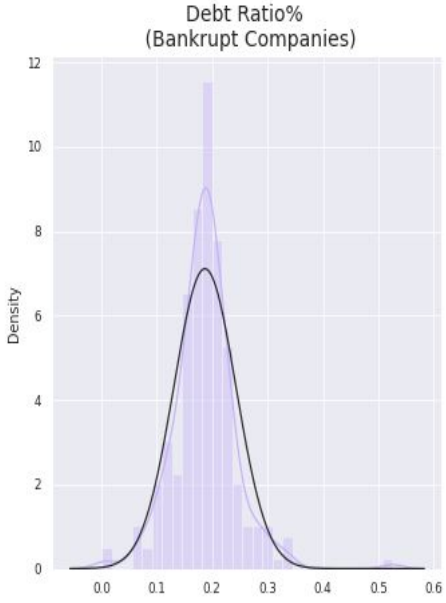


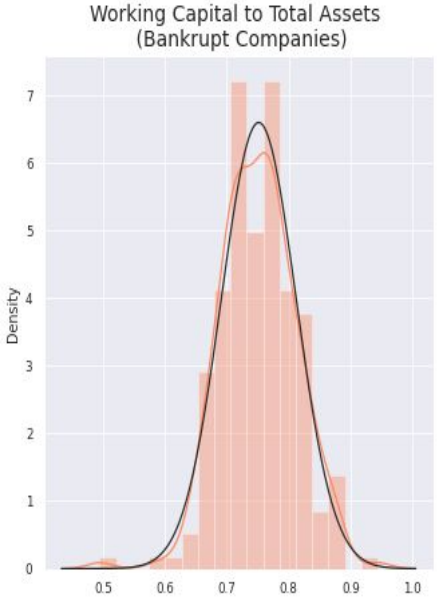


4. Plotting the feature distributions for close to bankruptcy companies(unstable Companies) after removing the outliers









5. Let’s Do the Log Transformation on our Data

- Log or Logarithmic Transformation is a data Transformation is a data transformation in which each variable is replaced with log of that variable
- Log Transformation reduces the skewness present in the data .
- It is done so that original data has to approximately follow Log-normal Distribution
- The Piece of code which does the log transformation is shown below:

```
def log_trans(data):

    for col in data:
        skew = data[col].skew()
        if skew > 0.5 or skew < -0.5:
            data[col] = np.log1p(data[col])
        else:
            continue

    return data

data_norm = log_trans(new_df)
```

AI

Data modelling

Data Modelling is the process of analyzing the data objects and their relationship to the other objects. It is used to analyze the data requirements that are required for the business processes.

Train test split

The procedure involves taking a dataset and dividing it into two subsets. The first subset is used to fit the model and is referred to as the training dataset. The second subset is not used to train the model; instead, the input element of the dataset is provided to the model, then predictions are made and compared to the expected values. This second dataset is referred to as the test dataset.

Lets implement Machine Learning Algorithms!

1. Logistic Regression :

- Logistic regression is a statistical model that in its basic form uses a logistic Function to model a binary dependent variable, although many more complex extensions exist. In regression analysis, logistic regression (or logit regression) is estimating the parameters of a logistic model (a form of binary regression).

- Mathematically, a binary logistic model has a dependent variable with two possible values, such as pass/fail which is represented by an indicator variable, where the two values are labeled "0" and "1".

Results from Logistic Regression:

All the Accuracy Scores are shown below in this picture. Which gives us Accuracy, Precision, Recall and F1-Score.

Classification report for Logistic Regression on the testing dataset

```
[ ]  print(classification_report(y_test, test_pred_lr, target_names=label))
```

	precision	recall	f1-score	support
Financially Stable Companies	0.99	0.83	0.90	1210
Financially Unstable Companies	0.14	0.77	0.24	44
accuracy			0.83	1254
macro avg	0.57	0.80	0.57	1254
weighted avg	0.96	0.83	0.88	1254

2. K-Nearest Neighbors (KNN) Classifier

- K nearest neighbors classifier is a non-parametric supervised learning algorithm first developed by Evelyn Fix and Joseph Hodges in 1951.
- It works by finding the distance between a query and all the examples in the data. For classification problems, a class label is assigned on the basis of a majority vote—i.e. the label that is most frequently represented around a given data point is used.
- To which example the query is closest to is calculated using different metrics such as Euclidean distance, Manhattan distance, Minkowski distance and other metrics.

Results from K-Nearest Neighbors Classifier:

Classification report for KNN classifier on the testing dataset

```
[ ]  print(classification_report(y_test, test_pred_knn, target_names=label))
```

	precision	recall	f1-score	support
Financially Stable Companies	0.97	0.86	0.91	1210
Financially Unstable Companies	0.07	0.27	0.11	44
accuracy			0.84	1254
macro avg	0.52	0.57	0.51	1254
weighted avg	0.94	0.84	0.88	1254

3. Naive Bayes Classifier (Gaussian)

- Naive Bayes classification is an algorithm that is based on the Bayes' Theorem. It is not a single algorithm but a family of algorithms where all of them share a common principle.
- Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.
- It is a probabilistic classifier, which means it predicts on the basis of the probability of an object. Naive Bayes classifier is also used for text classification algorithm.

Results from Naive Bayes (Gaussian)

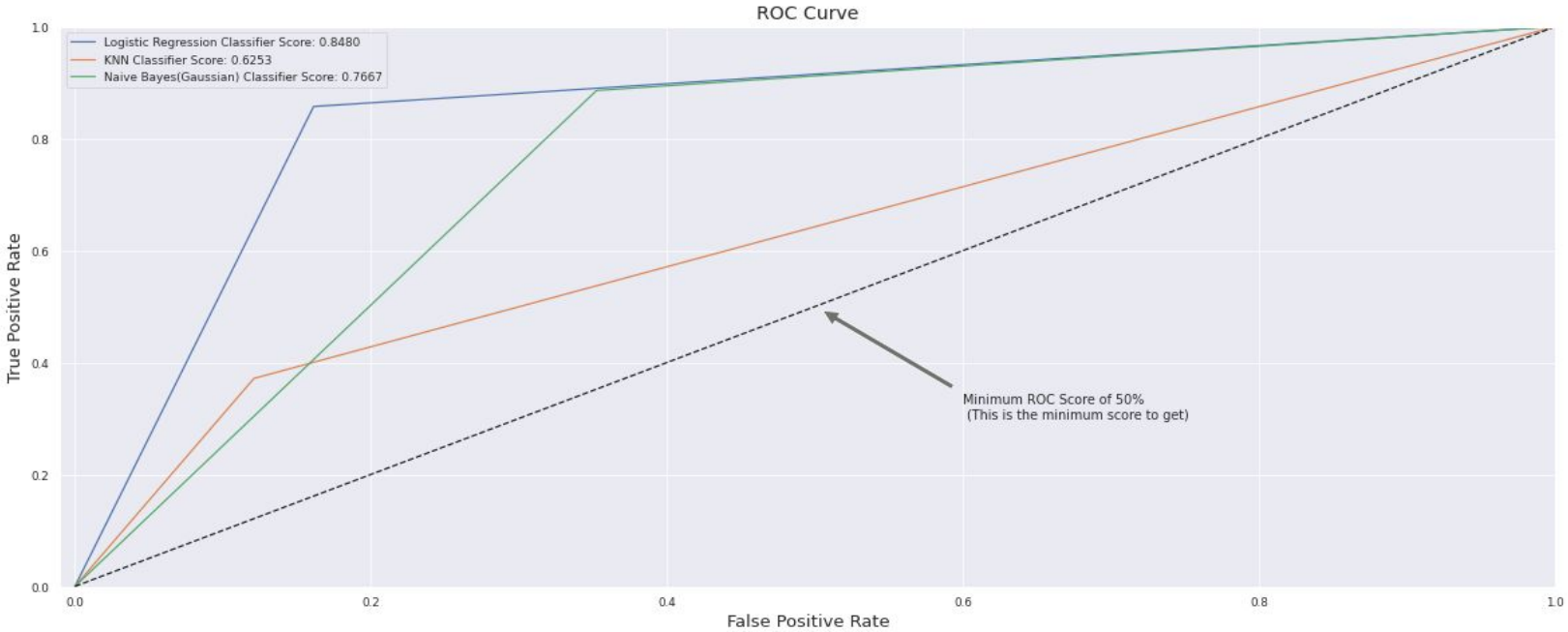
Classification report for Naive Bayes (Gaussian) on the testing dataset

```
[ ]  print(classification_report(y_test, test_pred_NB, target_names=label))
```

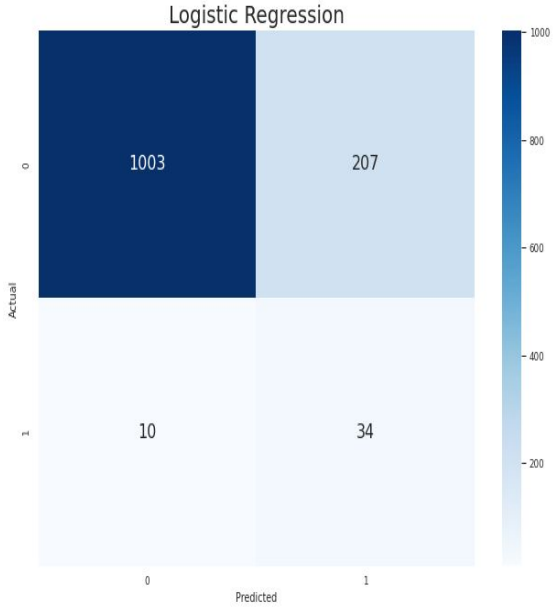
	precision	recall	f1-score	support
Financially Stable Companies	0.99	0.65	0.79	1210
Financially Unstable Companies	0.09	0.91	0.16	44
accuracy			0.66	1254
macro avg	0.54	0.78	0.47	1254
weighted avg	0.96	0.66	0.77	1254

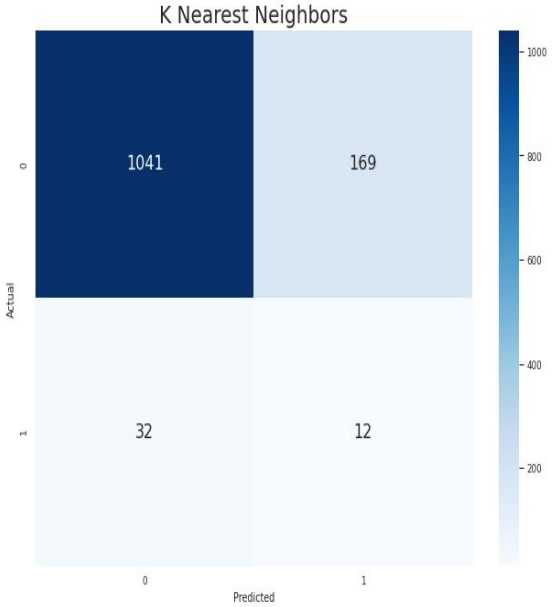
Plotting AUC-ROC curve

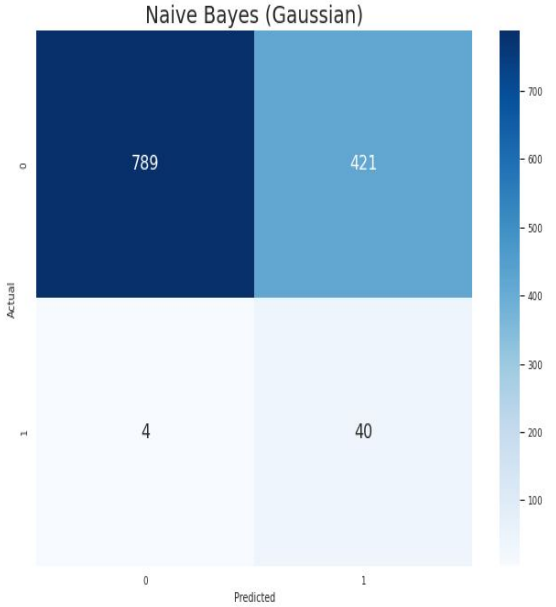
AUC - ROC curve is a performance measurement for the classification problems at various threshold settings. ROC is a probability curve and AUC represents the degree or measure of separability. It tells how much the model is capable of distinguishing between classes.



Confusion Matrix for each classifier (on the testing dataset)







Conclusion:

- The Experiment I chose is to predict if a company is going bankrupt according to the given features.
- The dataset has 65 features on offer .
- We applied 4 Machine Learning Algorithms namely :
  1. Logistic Regression
  2. K-Nearest Neighbors
  3. Naive Bayes (Gaussian)
- according to the f1-scores and the Accuracy metrics, KNN is the best metric to choose for further predictions.

- But according to AUC-ROC (Area Under the Curve – Receiver Operating Characteristics) says the larger the number, the greater the model.
- So, according to AUC-ROC, Logistic Regression stands first with a score of 0.84
- Nevertheless, in this case, the best decision is to use Logistic regression because it can better recognize the minority class even misclassifying some not close to bankruptcy companies as close to bankruptcy.
- Also the parameters that contribute to bankruptcy of a company are also external such as decisions taken by the CEO of the company and dwindling workforce and so on. Which can neither be calculated nor predicted.